Towards Multi-View Object Class Detection

Alexander Thomas, Vittorio Ferrari,
Bastian Leibe, Tinne Tuytelaars,
Bernt Schiele & Luc Van Gool

The Goal

Detection of specific objects

Figure credit: David Lowe

Recognizing classes of objects – typically limited to a single viewpoint

Multi-view object class detection

Multi-view specific object recognition system

V. Ferrari, T. Tuytelaars, and L. van Gool, Simultaneous Object Recognition and Segmentation by Image Exploration, ECCV, 2004.

V. Ferrari, T. Tuytelaars, and L. Van Gool, IntegratingMultiple Model Views for Object Recognition, CVPR, Vol. II, pp.105-112, 2004.

Implicit Shape Model for object class detection

B. Leibe and B. Schiele. Scale-Invariant Object Categorization using a ScaleAdaptive Mean-Shift Search, DAGM, pp. 145-153, 2004.

Image Exploration

Ferrari et al.

- Recognizing specific objects from different views
- Creating correspondences among different model views

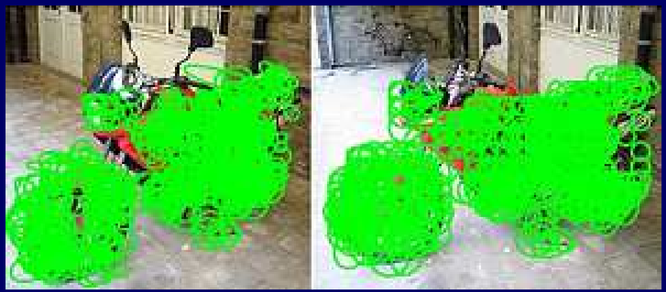
Region Tracks

A region track is composed of the image region of a single physical surface patch along the views.

A set of region tracks is produced for each specific training object.

Image Exploration

- Dense two-view matches are produced between each model image and all other images within a limited change of the view-point



- All the pairs of matches are integrated into a single multi-view model

Implicit Shape Model
Leibe & Schiele

- Recognizing object categories
- Codebook of local structures:
  - Clustered image features sampled at interest point locations
- Occurrences - map sampled image features from the test image to the codebook entries

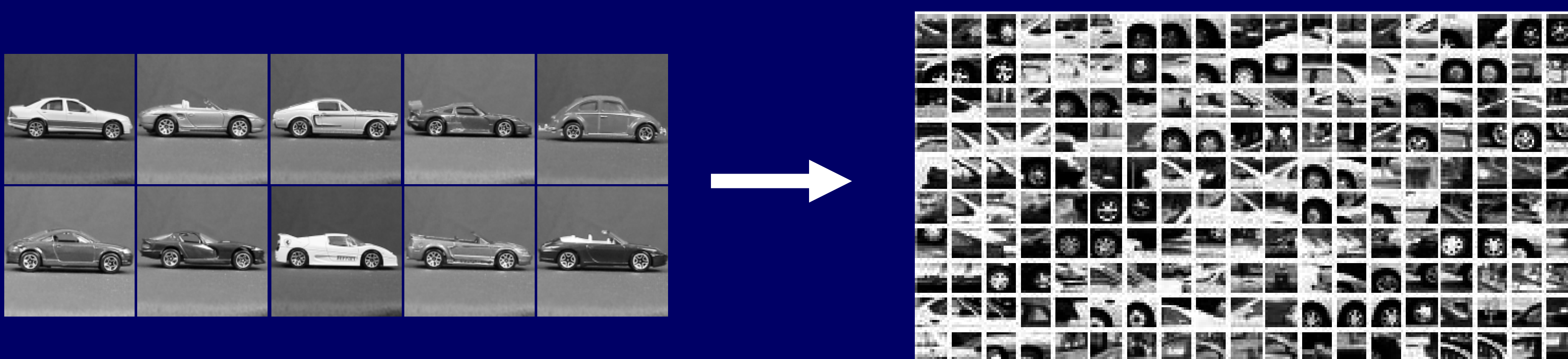


Image credit: Grauman & Leibe

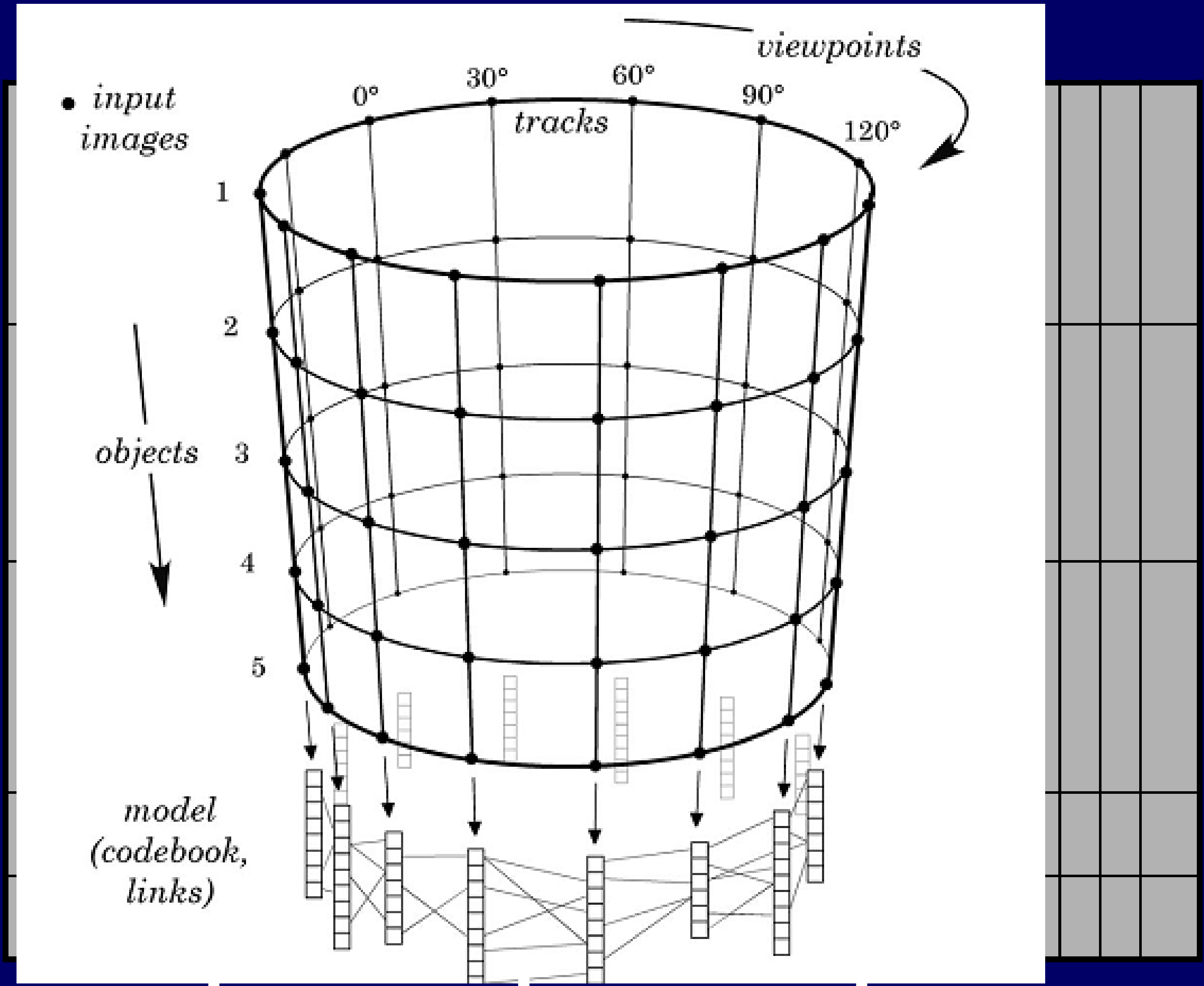
Integrating the Systems

- Integrate the two systems to achieve a multi view object class detection, not only by running a collection of single view detectors.
- The single view codebooks (ISM) will communicate using activation link (image exploration)

Training

- M object instances, from N viewpoints.
- The viewpoints should be approximately the same, but each instance does not need to have all of them.
- A set of ISMs is trained independently for each viewpoint.
- The image exploration algorithm is run for every object and create sets of region tracks.

The Data Set



Region Tracks

Region Tracks

Region Tracks

ISM

ISM

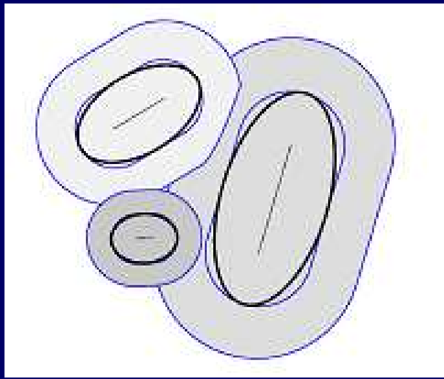
ISM

Region Tracks

- Region tracks – contain regions corresponding across the object's views.
- Region – described by ellipse – affine transformation of a circle.
- The affine transformation between the regions approximates the affine transformation between the image patches they cover.

How to find the closest region to the occurrence?

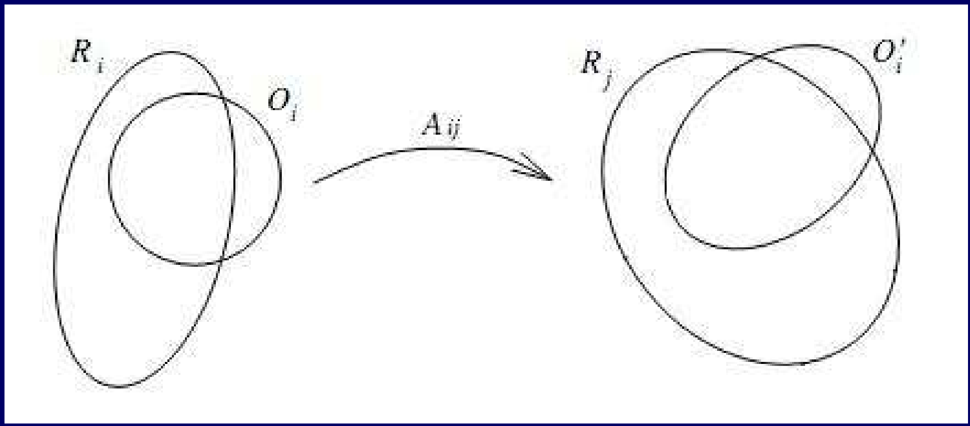
- There is no one to one correspondence between regions and occurrences.
- Finding the closest ellipse (a region) to a point (the center of the occurrence)
- There is an analytical solution, but it is computationally expansive.
- Approximation – the distance to a line aligned with the major axis of the ellipse, of length || l || - || s ||
- Only if the distance is < 2·||s||



Linking Algorithm

Iterate over all occurrences Oi in all training viewpoints of a specific object. For each Oi:

- Find the nearest **region** Ri (approximate way)
- For every other view Vj in Rj's track:
  - Transform the circular region Oi with affine transformation Aij (between Ri and Rj) to Oi'
  - Look for occurrences O_j^k in view Vj that are sufficiently similar to Oi'
  - Store all Oi -> O_j^k as activation links



Matching Occurrences

- Looking for all circles sufficiently similar to the ellipse Oi'.
- Using the heuristics:

(circle: pc – center, R – radius,
ellipse: pe – center, || l || || s || - major/minor axis)

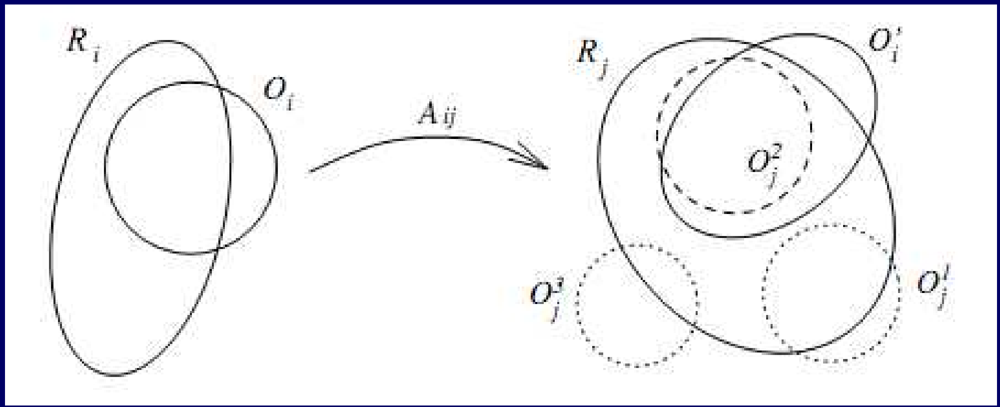
$|| p_c - p_e || < a \cdot R$

$| 1 - (||s|| \cdot || l ||)/R^2 | < b$

$|| s || / R > 1/c$

$|| l || / R < d$

a=0.35 b=0.25 c=d=3.0



Identification Algorithm

Handling a test image:

First stages – similar to ISM:

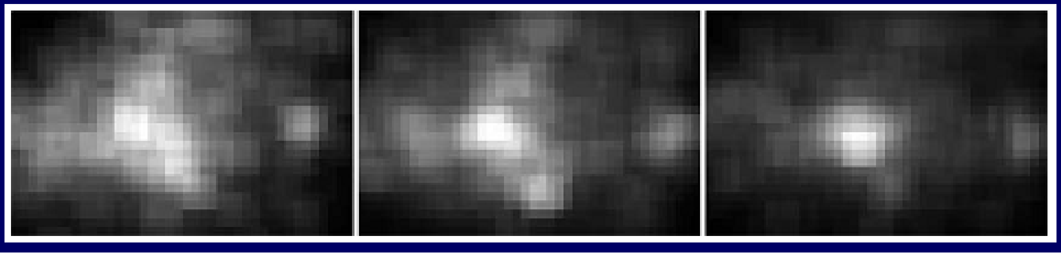
- Extracting features and matching to the codebooks of the different ISMs.
- Casting votes in the Hough spaces of each ISM separately.
- Detecting initial hypotheses as local maxima.

Identification Algorithm – Selecting Working Views

A trivial criterion – choose the views containing the strongest initial hypothesis

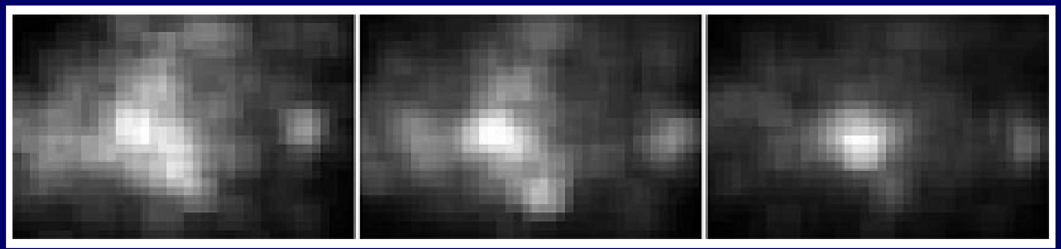
but…

- Image clutter can lead to strong hypotheses
- Correct strong hypothesis tend to create maxima in neighboring views while clutter doesn't.
- The pose of the object mostly falls between two training views.



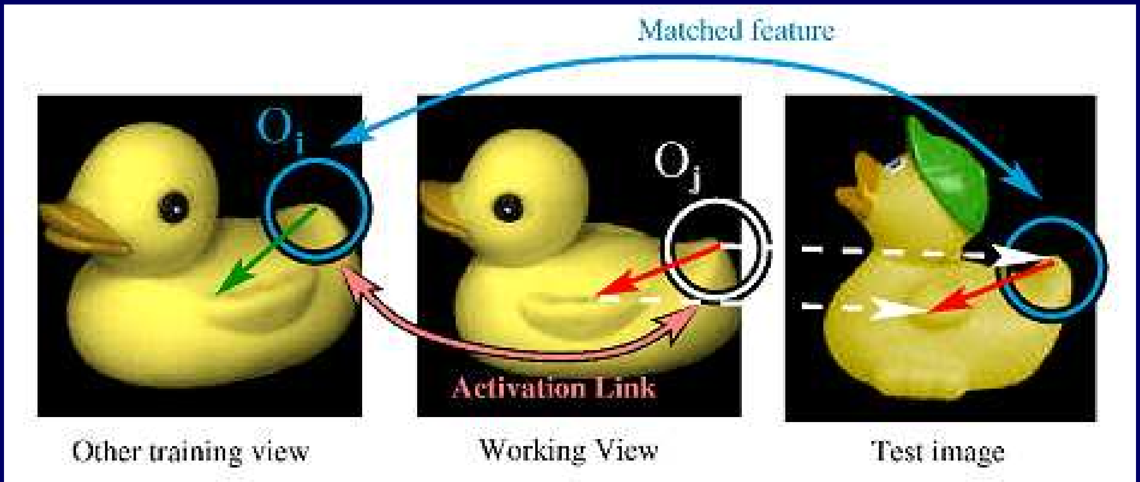
Identification Algorithm – Clustering the Hypothesis

- Pick the strongest hypothesis.
- Search in the neighboring views for hypothesis in approximately the same locations.
- Extend the cluster as possible to all directions.
- Take the next strong hypothesis etc. till all the hypotheses are clustered.
- The score of the cluster – the sum over all the hypotheses scores.
- Keep clusters > T * maximum score (T = 0.7).
- Choose the working views – the strongest hypothesis of each remaining cluster.



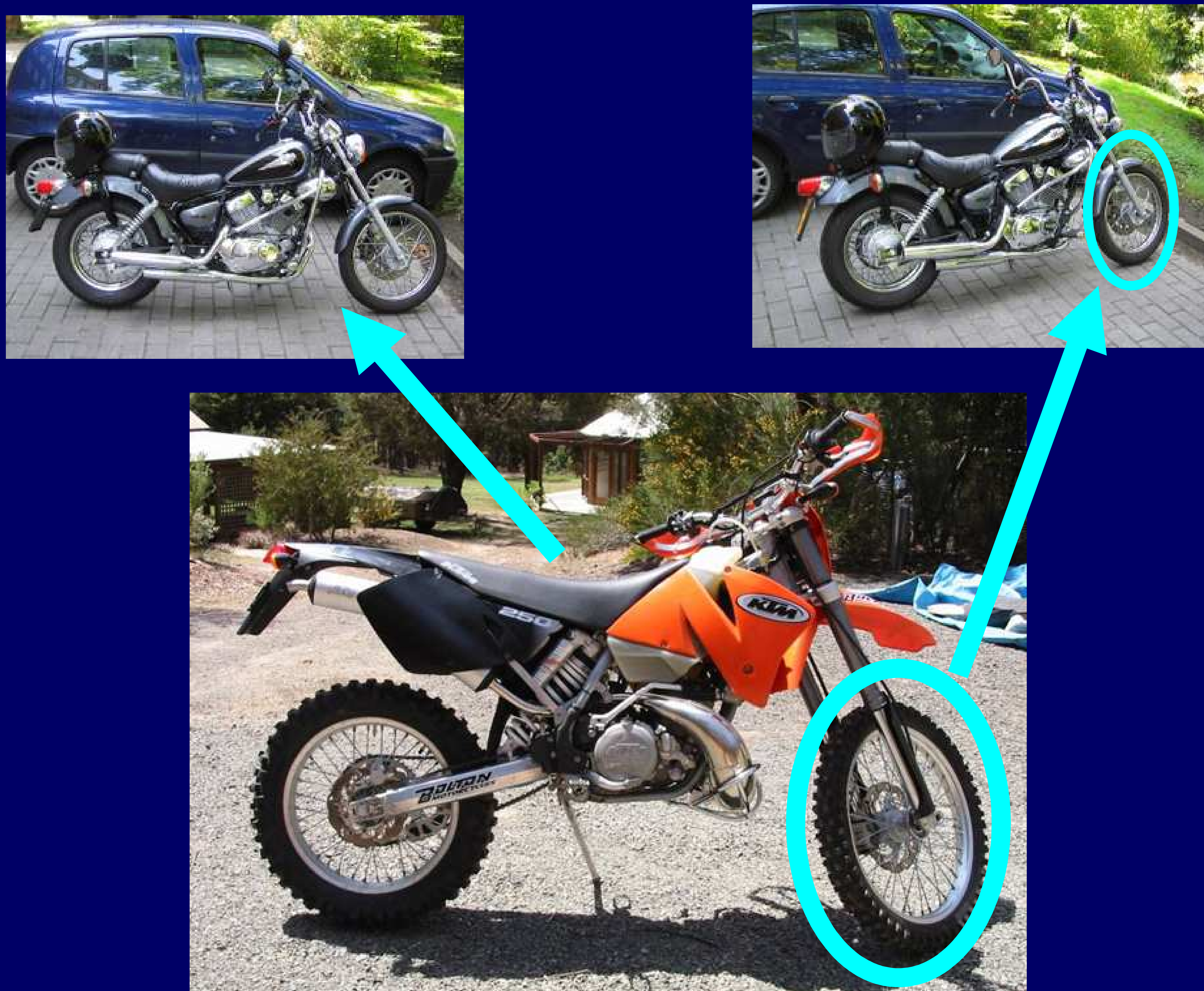
Identification Algorithm – Transferring Votes

Augmenting the Hough transform:



A feature matches to a codebook entry in view Vi + An activation link between entry's occurrences in Vi and Vj ⟶ Cast additional vote in Vj

- Assume that the part will be in approximately the same location in view Vj (for estimating the object center).
- If a part was detected in the codebook of Vi, but Vj is more likely the pose of the object, transfer the evidence of the part to Vj.


KTM
250
BOLTON

Identification Algorithm – Wight of Transferred votes

Expresses the contribution of a patch e in location l to an object hypothesis (o_n,λ) (λ - x y s – location and scale). Vj is the current working view.

$$p(o_n, \lambda | e, \ell) \;=\; \sum_k P(o_n, \lambda | c_k^j, \ell) p(c_k^j | e) + \sum_k \sum_l P(o_n, \lambda | c_k^j, c_l^i, \ell) p(c_l^i | e)$$

Iterates over all the entries for view Vj

The probability of the hypothesis given the codebook entry

The probability that the entry C_k in view Vj is a correct interpretation of the patch

Iterates over all the entries of the other codebooks – Vi ≠ Vj

Non-zero only if there is an activation link between c^i_l and c^j_k

The probability that the patch matches the codebook entry C_l in view Vi

Testing

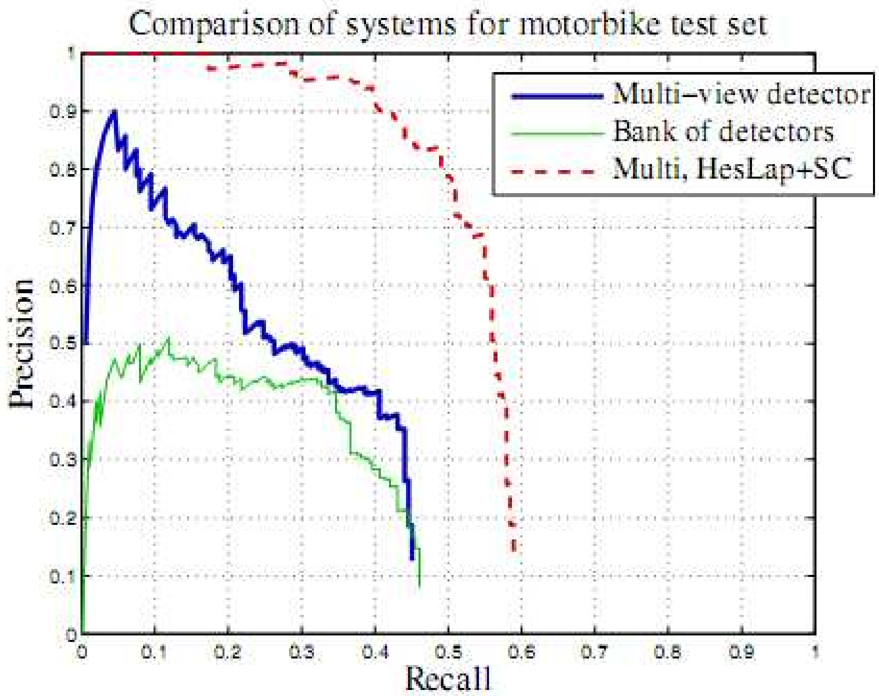
- Motorbikes from PASCAL Visual Object Classes (VOC) Challenge and sport shoes.
- Motorbikes - training set - 30 objects, segmented by a bounding box, 16 training views taken on a circle around the object. Average of 11 views per motorbike.
- Average of 22 objects per view point, which is only a small number for training the ISM.
- Sport shoes - training set - 16 views, taken at 2 different elevations.

  Test set – collected form Google, Flickr and Fotolog.com .

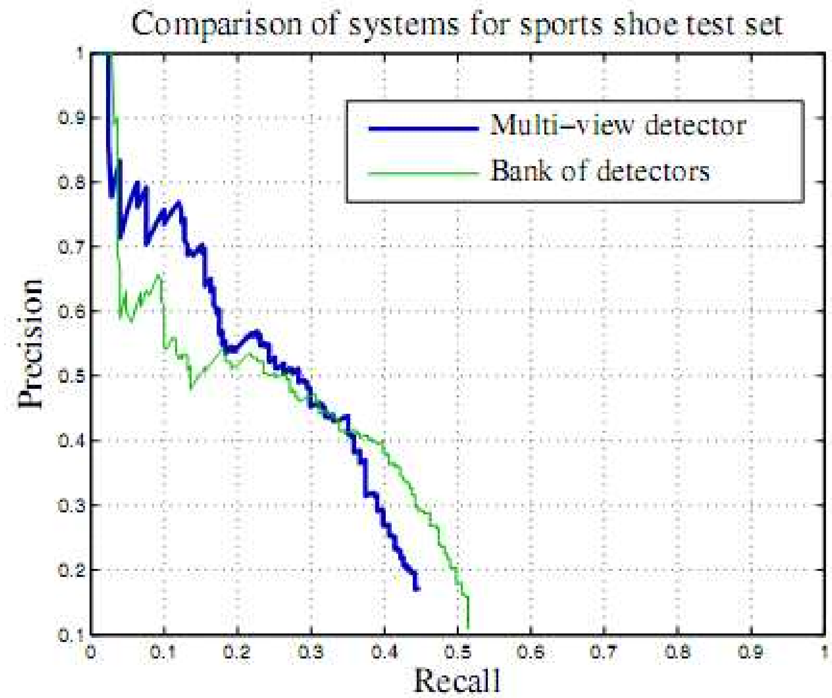
Testing

- Baseline: a bank of 16 ISM models ran separately.
- All the detections are collected and output together.
- Evaluation protocol like in the PASCAL challenge – detection is correct if its bounding box overlaps more than 50% with the ground truth.

Testing



Comparison of systems for motorbike test set
Multi–view detector
Bank of detectors
Multi, HesLap+SC
Precision
Recall



Comparison of systems for sports shoe test set
Multi–view detector
Bank of detectors
Precision
Recall

Testing

- Comparison versus PASCAL VOC challenge – second using DoG+Patches and first using the new HesLap+SC features.
- After training the ISMs from much fewer motorbike instances.
- Not a perfect comparison:
  - Trained on different instances
  - Used multiple training views per instance

Results

Multi View System

Bank of Detectors